International Journal of Pure and Applied Mathematics

Volume 2 No. 4 2002, 439-486

NEURAL MODELS OF CHAOTIC ATTRACTORS AND CHAOTIC FEATURES OF NEURAL NETWORKS

Nikos Kofidis[1], Manos Roumeliotis[2 §], Miltiadis Adamopoulos[3]

Dept. of Applied Informatics
Economics and Social Sciences
University of Macedonia
P.O. Box 1591, Egnatia 156
Thessaloniki 54006, GREECE

[1]e-mail: kofid@uom.gr
[2]e-mail: manos@uom.gr
[3]e-mail: miltos@uom.gr

Abstract: This paper investigates the ability of neural structures to model chaotic attractors and the chaotic features of neural networks. The objects of modeling are the logistic and Henon attractors. A significant improvement of model behavior is achieved when "multiple training" is applied. The improved model consists of a complete set of submodels, that is, networks which simulate a part of the attractor, driven by an LVQ controller. The series of absolute errors produced during the recall phase of the best neural models of each map is submitted to Lyapunov exponent investigation. The chaotic behavior of the absolute error that emanates from a single step prediction of the chaotic orbits indicates that neural networks, when fed only with an initial input, are able to produce the geometrical object corresponding to the simulated attractor. However, they fail to follow the original chaotic path (orbit) of the attractor. The chaotic features of the training phase are examined by investigating the dominant Lyapunov exponents of the weights series, and series of input patterns producing the minimum absolute error in the output.

Received: May 15, 2002

§Correspondence author

The best estimation series, consisting of the output values corresponding to the best-learned input patterns, are also examined, using the theoretical tool of topological conjugacy.

AMS Subject Classification: 65P20
Key Words: chaos theory and models, neural networks, nonlinear dynamical systems

1. Introduction

1.1. The Logistic and the Henon maps

The one-parameter family of logistic maps is described by the equation $X_{n+1} = \lambda X_n(1 - X_n)$. When $\lambda > 3$, the period doubling route to chaos, known as Feigenbaum senario, starts. It has been proven [8], that there exists a value of $\lambda = \lambda^*$, which fires chaotic behavior. In this paper, the data used to train and check the neural model of the map have been taken for $\lambda = 3.93$. This value lies somewhere in the middle between interior ($\lambda = 3.857$) and boundary crises ($\lambda = 4$) ([3], [12]).

The Henon map is a two-parameter two-dimensional map [11], described by the equations $X_{n+1} = 1 + Y_n - aX_n^2$ and $Y_{n+1} = bX_n$. It has been shown that the parameter values a=1.4, b=0.3 produce a strange attractor. This study considers these parameter values together with an initial vector $(X_0, Y_0) = (0, 0)$, to produce training and testing data sets.

1.2. Chaotic Behavior and Lyapunov Exponents

Chaotic behavior is a common feature of many dynamical systems. It emanates from nonlinearity and complexity, and is primarily investigated by examining the essential property of sensitivity to initial conditions, which is measurable through the Lyapunov exponents. These exponents represent the average exponential rate of expansion (contraction) of the axis of the ellipsoid produced, when the measured system evolves acting on a unit sphere of initial conditions. The eventual zero volume of the ellipsoid in dissipative systems presumes at least one negative Lyapunov exponent, meaning that expansion is not possible in all directions of the N-space. Although there may exist more than one positive Lyapunov exponents, representing chaotic response in an N-dimensional system, the dominant Lyapunov exponent, corresponding to the most expansive axis of the ellipsoid, is the governing one. Schuster [20], Alligood [2], and

Hilborn [12] have described methods for calculating this exponent, for systems with known evolution equations.

The estimation of Lyapunov exponents from experimental data seems to be more complicated. J. Wright [26], Allan Wolf, J.B. Swift, H.L. Swinney, and J.A. Vastano [25], J.P Eckman and R. Oliffson, S. Kamphorst D. Ruelle, and S. Ciliberto [7] have proposed various algorithms.

Ordinary neural network structures are nonlinear, complex dynamical systems. Although convergence is a desirable property, and almost always the task for many researchers, it seems to be an exceptional functional status, since chaotic regime is present even in small sized neural networks.

2. Research Tasks and Previous Work

2.1. Overview

This paper consists of two parts. The first part (Sections 3 and 4) examines the ability of Back Propagation neural networks to model the chaotic attractors of the logistic and the Henon maps. The construction of neural models of the attractors is based on the following principles:

- Structural uniformity, meaning that a general structure should be proposed to satisfy modelling requirements for both maps.
- The RMS and Average errors are not considered satisfactory criteria for the networks' behavior. Additional calculation of the distribution of the absolute error over pre-determined error levels is necessary for a more reliable picture of the networks' response.
- Functional Link (FLN) inputs are used.

Single training is used for the construction of the basic models, while multiple training is used to improve the network' behavior. During single training, the networks are trained to represent the whole attractor. In this case, the training data set is representative of the whole input space of the attractor. When multiple training is used, the network is trained to a subinterval of the input space of the attractor. This network, called a submodel, behaves much better than the basic model, when the chaotic orbit moves in the part of the attractor that it represents. A problem is imposed by the fact that none of these submodels can function independently. Due to the property of topological transitivity of the chaotic systems, no functionally independent subsystem

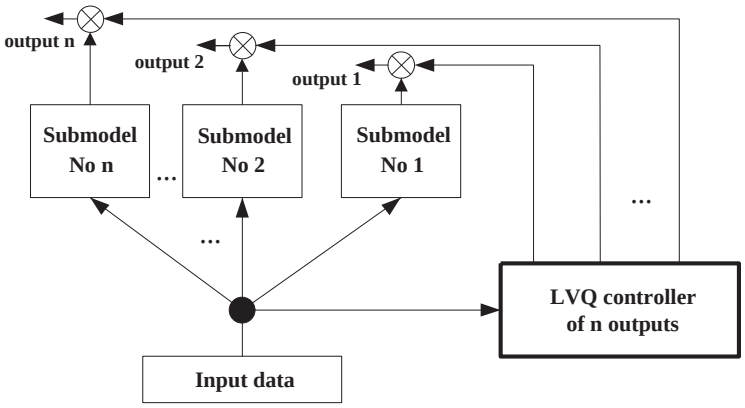

Figure 1: A complete set of submodels driven by an LVQ controller

exists. More precisely, $f : A \to A$ is said to be topologically transitive if for any pair of open sets $U, V \subset A$ there exists $k > 0$ such that $f^k(U) \bigcap V \neq \emptyset$ meaning that any input value leads to any part of the attractor after a finite number of iterations. The above considerations make it clear that the set of submodels covering the whole attractor should work together in a competitive manner, if a reliable model for the attractor is required. LVQ networks are proposed to solve this problem. As shown in Figure 1, an LVQ network acts as a signal controller, activating the output of the proper submodel. The LVQ network has as many inputs as the models (one for the logistic and two for the Henon map model) and as many outputs as the number of subspaces in which the input space is divided.

The second part (Sections 5 and 6) explores the existence of chaotic features in specific neural structures, and its effects.

In Section 5, the absolute error series produced during the recall phase of the best basic models of the two maps are submitted to a Lyapunov exponent analysis. Its chaotic response sets limits to the attractors' neural modelling.

In Section 6, simple converging or non-converging Back Propagation neural networks are examined during the learning phase, to investigate chaotic properties. The most interesting point of this investigation is the pattern competition emanating from the study of the movement of the input value, corresponding to the active output with the minimum error. Topological conjugacy and numerical verification are the main tools of this investigation.

2.2. Previous Work

Several researchers have worked on modelling of chaotic attractors and the exploration of chaotic features of neural networks.

A. Lapedes and R. Farber [16] constructed Back Propagation neural networks to predict chaotic time series, using the appropriate number of past system values to feed the input module. On the other hand, S.F. Masri, A.G. Chassiakos and T.K. Caughey [17] have investigated the ability of Back Propagation networks with static or dynamic neurons to learn non-linear dynamical systems. Small changes of parameter values were allowed, while experimental measurements were used to provide the network with the displacement and velocity values of the dynamical system.

As far as widely known attractors are concerned, George J. Mpitsos and Robert M. Burton. Jr [18] studied in detail the effect of input signal dynamics to the neural network learning process, by modelling the logistic map for $\lambda = 3.95$, random noise, and *sine* functions, with Back Propagation neural networks, while Ramazan Gencay [9] constructed feed forward networks to predict non linear time series produced by the Henon map application when noise is present. A more recent work has been presented by I-Cheng Yeh [13], who constructed Back Propagation neural networks with added extended layer and auxiliary output neurons to model the Henon and Ikeda maps.

Steve Renals and Richard Rohwer [21] investigated the dynamics of discrete neural networks consisting of N sigmoid nodes fully connected via non-symmetric (in general) weight matrix. Considering the slope of the transfer function "r" and the weight matrix symmetry parameter "a" as the varying parameters of the system, they systematically exhibited, calculating power spectra for each node, the rich dynamical behavior of networks consisting of 2 to 25 nodes. For a specific 8 node-network, a more detailed picture of chaotic behavior and routes to chaos were presented via bifurcation diagrams.

John F. Kolen and Jordan B. Pollack [15] have experimented on simple Back Propagation networks, to indicate sensitivity of convergence on initial choice weights, learning rate, and momentum. More specifically, they constructed 2-2-1 networks, and let one pair of weights to vary each time, in order to find out regions of initial weight values that lead or not to t-convergence (convergence after t epochs of training). The same method was applied to 3-3-1 networks, to determine the effect of varying learning rate and momentum in t-convergence.

Han L.J. Van der Maas, Paul F.M.J. Verschure and Peter C.M. Molenaar [23] have explored chaotic behavior of the sum of the weights of a 3-cell autoassosiator, trained by the use of Back Propagation and Hebian rules. Bifurcation

diagrams, Lyapunov exponents, and Power spectra indicate chaos in both cases when the learning rate, playing the role of control parameter, takes its values from certain ranges.

Thomas B. Kepler, Sumeet Datt, Robert B. Meyer and L.F. Abbot [14] implemented a four node neural network circuit in order to explore possible pathways to chaotic behavior. As shown period doubling, intermitency, and quasi-periodic pathways are all possible roots to chaos for neural networks.

Paul F.M.J. Verschure [24] has utilized the neural networks' chaotic behavior to create a chaos-based learning algorithm according to which the network after the stabilization of the error function is driven to chaos through a step by step increase of the learning rate. Then, a cooling down phase follows, based on a Metropolis like algorithm, until final network stabilization.

Francois Chapeau-Blondeau and Gilbert Chauvet [5], studied the behavior of two and three node neural networks with full or partial delay. Chaotic response was clearly depicted via bifurcation diagrams and Lyapunov exponent estimation for the 3-node network with partial delay. Weight w13 was the varying parameter, while in the full delay case, the variation of the sigmoidal slope lead to fractal structure presentation of the output.

K. Aihara, T. Takabe and M. Toyoda [1] proposed a neuron model with chaotic dynamics, based on the Nagumo-Sato model. Neural networks consisting of such neurons, namely chaotic neural networks, include conventional neural networks as a special case (McCulloch-Pitts and Back Propagation).

E.K. Blum and Xin Wang [4] explored the dynamics of small neural networks of the sigmoidal type, using the corresponding difference equations, under the consideration of time discreteness and synchronization.

G. Randons, H. G. Schuster and D. Werner [19] considered the iterational procedure of weight updating, with learning rate as the varying parameter. This resulted in fractal measures (Cantor sets) for the invariant distributions of the weights P(w).

3. The Logistic Map Neural Model

3.1. Network Architecture

The task of modelling the logistic map chaotic attractor led, after several tries with various structures, to a Back Propagation network with one hidden layer and Functional Link (from now FLN) inputs. The same architecture is also used for the Henon attractor neural model, satisfying the demand of uniformity

of the neural models. In the following, a general network architecture of the logistic map model is described. From that general architecture (Figure 2), emerge the specific neural models by changing:

a) the FLN input structure,

b) the transfer function of the hidden layer and

c) the transfer function of bypass neuron (when it is present in the structure).

Input modules: The main input module consists of one cell fed with the X_n value of the logistic map. Two types of FLN modules are used: simple and extended. Simple FLN feeds the hidden layer with the main input and with the four modules of FLN-Sk, $k = 1, 2, 3, 4$, as given in Table 1. Extended FLN applies to the hidden layer the main input, the FLN-Sk modules and extra information produced in the FLN-EXk modules, also given in Table 1.

Name of Input Module	Number of cells	Cell Input	Cell Output
Main Input	1	X_n	X_n
$FLN - S_k$	4	X_n	$sin(k\pi X_n), k = 1, 2, 3, 4$
$FLN - E_k$	4	X_n	$X_n sin(k\pi X_n), k = 1, 2, 3, 4$

Table 1: Input modules

Hidden layer: As realized after a few tests, the simplest and most efficient architecture is the one using a single hidden layer consisting of three neurons. A second layer does not improve the performance of the network (see also [18]). On the other hand, adding more neurons to the single hidden layer results in an increased complexity of the system, which degrades the response of the network in some cases (see also [18]). The hidden layer is fully connected to the output layer with trainable weights and, for the best models, it applies the *tanh* transfer function. Each neuron of the hidden layer is connected to a trainable BIAS neuron.

Output layer: The output layer consists of one neuron. This single neuron is responsible for producing the X_{n+1} step of the logistic map, by linearly combining the outputs of the hidden layer and the trainable BIAS.

Bypass neuron: This is a single neuron layer that is fed through trainable weights with the outputs of the hidden layer. Its output is connected to the output layer of the network. The weights used for the connection of the bypass neuron to the output are also trainable. The bypass neuron does not prevent the direct flow of information from the hidden to the output, as shown in Figure 2. The *sine* function is used as a transfer function of the unit, in the best

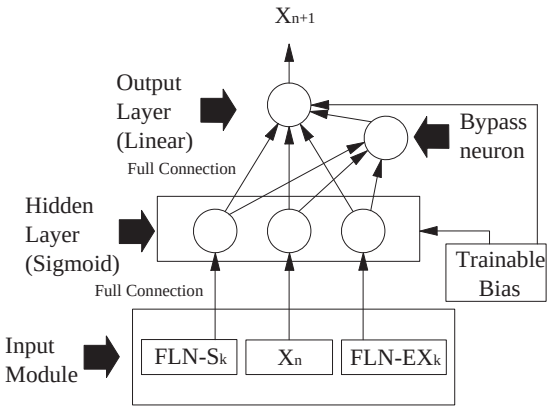

Figure 2: FLN-Back Propagation Network that models the logistic map

responding networks.

3.2. Parameters of Training and Testing

The Back Propagation delta rule algorithm was used to train the networks. The learning rate was set at 0.1 and the momentum at 0.85. This combination of values gave the best results for fast convergence [10]. The initial weights were limited within [-0.1, 0.1]. The epoch was set to 10 and the maximum number of iterations was kept fixed at 30000 (3000 epochs). The data sets used for training consisted of 500 input-output pairs, produced from the logistic map for the parameter value $\lambda = 3.93$. This value of stands in the middle between interior and boundary crises. The input-output pairs do not correspond to consecutive points on the chaotic orbit but were sampled uniformly from the reference space. On the other hand, the testing set consisted of 8000 input-output pairs corresponding to consecutive points on the chaotic orbit emerging from an initial value equal to 0.1. In order to check the stability of the networks' response to the parameter fluctuation, 15 data sets were created each one corresponding to one value of $\lambda \in [3.86, 4]$ for the logistic map. The interval between consecutive $\lambda - values$ was kept to 0.01. Table 2 summarizes the parameter values used in training and testing mode.

To check the networks' performance, Average, RMS and distribution of absolute errors were taken in to consideration. The absolute error, E_{abs}, is calculated as the absolute value of the difference between the actual output value produced by the network and the desired one:

Type of Network	Back Propagation network with functional link inputs
Learning Rate	0.1
Momentum	0.85
Range of initial weight space	[-0.1, 0.1]
Data set used for training	500 pairs uniformly distributed
Epochs	10
Maximum number of iterations	30000 (3000 epochs)
Data sets used for testing	8000 pairs corresponding to consecutive points on the chaotic orbit with initial value 0.1

Table 2: Parameters of training and testing

$$E_{abs} = |X_{out} - X_{des}|, \tag{1}$$

where X_{out} is the actual and X_{des} is the desired output of the network. From the distribution of the absolute error over pre-chosen error levels, a statistical overview of the networks' failure emanates. More specifically, for a given error level questions like "what is the percentage of output data with absolute error greater than" can be answered.

3.3. Single Training

During single training, the networks are trained to the whole input space, to generalize the logistic map attractor for $\lambda = 3.93$. Table 3 summarizes the features of the neural models with the best performance during training. Each of the networks corresponds to a certain structural aspect.

Hidden Layer	Bypass Neuron	Type of FLN	RMS of the data set
tanh	none	simple	0.0017
tanh	sigmoid	simple	0.0013
tanh	none	extended	0.00051
tanh	*sine*	extended	0.00063

Table 3: The best networks during training by structure category

As expected, the best responding networks make use of the extended FLN input structure. These two networks have been thoroughly examined during the testing mode. Table 4 presents the RMS error, the Average error and the distribution of absolute error for the two networks with extended FLN inputs of Table 3. Although the RMS error of the testing set is slightly higher in the network without a bypass neuron, this network can be considered the best model as far as the parameter λ is kept fixed at the value 3.93. This consideration is based on the detailed picture of its performance given by the distribution of the absolute error.

The response of the best two networks to the fluctuation of the parameter is presented in the diagrams of Figures 3, 4, 5 and 6. As shown in Figure 3 and Figure 4, the network with the *sine* bypass neuron is more stable in parameter fluctuation. This consideration is based on the "0.005" error line, which is zero in the intervals [3.92, 3.94] and [3.92, 3.95] for the two types of networks.

Error Type	Extended FLN No Bypass	Extended FLN Sine Bypass
Average	0.00045	0.00048
RMS	0.00064	0.00063
$\%Error > 0.1$	0%	0%
$\%Error > 0.05$	0%	0%
$\%Error > 0.01$	0%	0%
$\%Error > 0.005$	0%	0%
$\%Error > 0.001$	6.79%	13.75%
$\%Error > 0.0005$	33.33%	40.51%

Table 4: The best two networks during testing phase

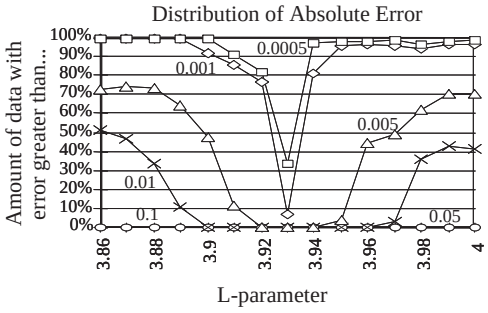

Figure 3: Extended FLN. No by pass neuron is used

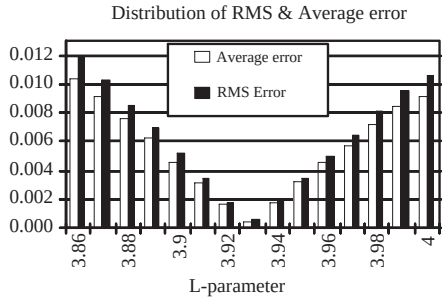

Figure 4: Extended FLN. No by pass neuron is used

3.4. Multiple Training

During multiple training, the network without a bypass neuron was trained to learn the map when input X_n moves in certain subintervals of [0,1]. Two cases were studied:

a) $X_n \in [0, 0.5]$ or $X_n \in (0.5, 1]$. This case corresponds to Figure 1 when $n = 2$ is applied. Two identical networks (of the above described structure) were trained, the first to memorize the map when the input interval is [0, 0.5] and the second when the input interval is (0.5,1]. The individual RMS error of each of the above networks during training is presented in Table 5 along with the average error.

The whole system of the two networks is driven by an LVQ network with one input neuron and two output neurons. In the Kohonen layer, LVQ applies one neuron per class. LVQ learns to recognize which of the above two subintervals

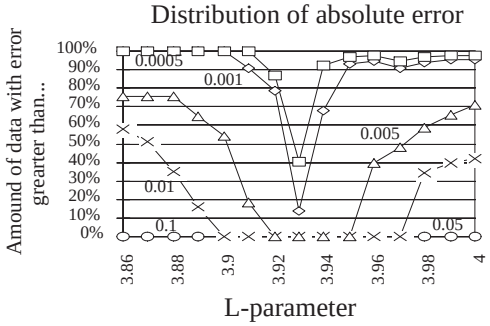

Figure 5: Extended FLN. *Sine* by pass neuron is used

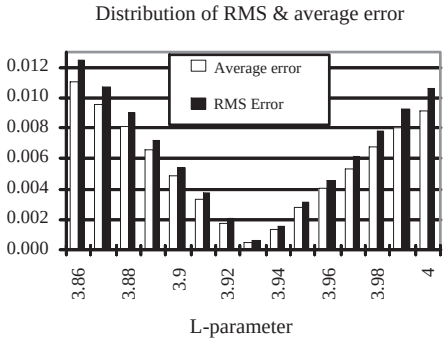

Figure 6: Extended FLN. *Sine* by pass neuron is used

	$X_n \in [0, 0.5]$	$X_n \in (0.5, 1]$
Average Error	0.00031	0.00024
RMS Error	0.00038	0.00034

Table 5: Corresponding error values of the data sets

the input X_n belongs to, and activates the corresponding output. The structure of the LVQ network used to control the system of the two competing submodels of the logistic map is shown in Figure 7.

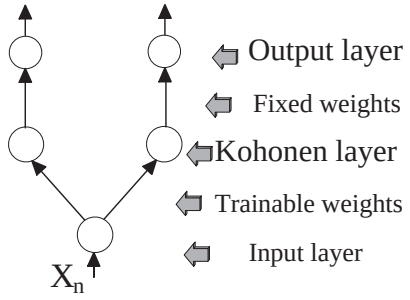

Figure 7: Two output LVQ network

b) $X_n \in [0, 0.3]$ or $X_n \in (0.3, 0.7]$ or $X_n \in (0.7, 1]$. This case corresponds to Figure 1 when n=3 is applied. Three identical networks are trained, each one learning the map for one of the input subintervals in discussion. The individual errors of the data set for each network are given in Table 6.

Now, the controlling LVQ network has three output neurons, each one activated when X_n belongs to the corresponding input subinterval. Again, one

	$X_n \in [0, 0.3]$	$X_n \in (0.3, 0.7]$	$X_n \in (0.7, 1]$
Average Error	0.00008	0.00010	0.0016
RMS Error	0.00013	0.00013	0.00019

Table 6: Corresponding error values of the data sets

neuron per class is used in the Kohonen layer. Figure 8 shows the structure of the LVQ network used in this case.

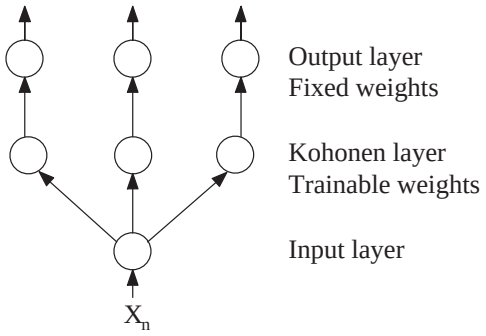

Figure 8: Three output LVQ network

The results taken during the testing phase for the three models of the logistic map are summarized in Table 7 and Figures 9 and 10. In these figures, n=1 corresponds to the absence of an LVQ controller.

	No LVQ controller	2 output LVQ controller	3 output LVQ controller
Average Error	0.00045	0.00028	0.0013
RMS Error	0.00064	0.00034	0.00017
$\%Err > 0.005$	0%	0%	0%
$\%Err > 0.001$	6.79%	0%	0.32%
$\%Err > 0.0005$	33.33%	12.07%	0.65%

Table 7: Comparative results for the logistic map models

As can be observed, the Average and the RMS errors are almost divided by two or three when the input space is divided in two or three (not necessarily equal) subintervals. More impressive is the picture of the absolute error for the error levels shown in Table 7. It is obvious that, for the last and more complicated system, almost every actual output is produced with error less

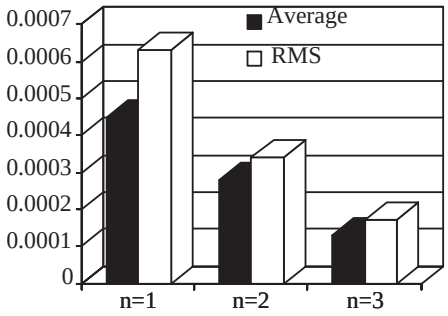

Figure 9: RMS and average error of the testing set for single and multiple training

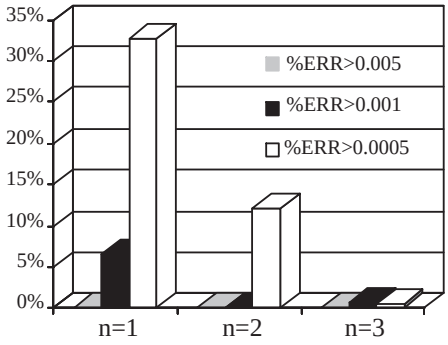

Figure 10: Distribution of the absolute error of the testing set for single and multiple training

than 0.0005. This is a satisfactory response, when low depth prediction is applied, and consequently, the results are not sensitive to initial conditions.

4. The Henon Map Neural Model

4.1. Introduction

Although most of the current research on strange attractors regards the Henon map as a two-dimensional map, here we consider it as a one-dimension map with double recursion. Accordingly, the neural structure has to memorize the formula $X_{n+1} = 1 + 0.3X_n - 1 - 1.4X_n^2$. This transformation results in a network with only one output neuron, which leads to lower system complexity and a significant reduction of the computational time.

It should also be mentioned that, all data used to train and test networks are normalized to the interval [0, 1]. Figures 11a and 11b show the original and normalized Henon attractors.

4.2. Network Architecture

The Henon map neural model retains the same architecture as the logistic map model. As shown in Figure 12, all structural units of the logistic map model are kept unchanged. The differences are restricted to the necessary addition of a second main input cell directly connected to the hidden and to the output

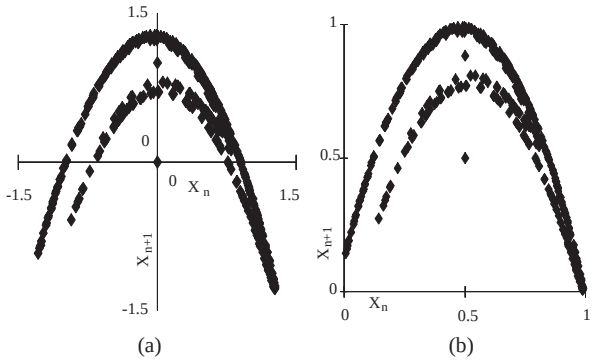

Figure 11: (a) Original and (b) normalized Henon attractor

layers ([16], [18]), with trainable weights. No FLN technique is applied to the network for this second unit of the main input.

4.3. Parameters of Training and Testing

The Back Propagation-delta rule algorithm was used to train the networks modelling the Henon map. The learning rate, momentum, epoch, range of the initial weight space and the maximum number of iterations were kept fixed. Their values were the same as for the logistic map and are given in Table 2. Training data sets consist of normalized data and correspond to points of the chaotic orbit starting at $X_0 = 0$, $Y_0 = 0$. In the case of single training, the data set consists of 500 input-output pairs corresponding to consecutive orbit points. However, when the networks are trained to submodels, this is impossible since consecutive points do not always satisfy the conditions, which the submodel input space obey by definition. The points corresponding to a submodel training set are consecutive in a new manner. It is considered that a point is next to another, when it belongs to both the chaotic orbit and the specific subspace of the inputs used for training the submodel.

The testing set is the same for all cases. It consists of 8000 normalized pairs of input-output values corresponding to consecutive points of the chaotic orbit mentioned before.

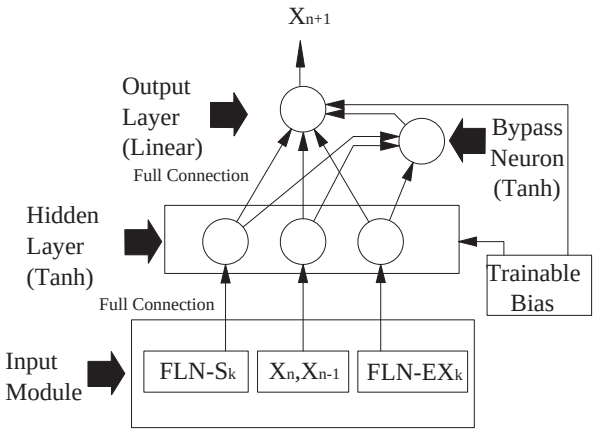

Figure 12: FLN-Back Propagation network that models Henon map

4.4. Single Training

A variety of structures, based on the general prototype described in Figure 12, emanate from altering the transfer function of the hidden layer and the bypass neuron, if it exists. Table 8 presents 12 networks and their response during training. As can be seen, the network with *tanh* transfer function in the hidden layer and *sigmoid* in the bypass neuron is the best responding one during training. Table 9 presents the detailed results produced during the testing phase, for the models of the Henon map, which gave the best response during the training phase.

Again, the *tanh-sigmoid* network is the best responding one, with regard to the RMS error and the absolute error level *0.001*. Although the *sine-none* network is better performing with regard to error level *0.0005*, it cannot be considered as the best model because of the diffusion of a significant part the observed error above the *0.001* level. Consequently, the *tanh-sigmoid* network is selected to support multiple training neural models of the Henon map.

4.5. Multiple Training

The double recursion of the Henon map implies a two-dimensional input space, thus increasing exponentially to the number of subintervals of [0, 1], the number of submodels needed to form a complete modelling set. For example, when [0,1] is divided in two subintervals, four networks are to be trained as submodels. If [0, 1] is divided in three subintervals, nine networks will be needed, and so

	Transfer Function		Error	
	Hidden Layer	Bypass Neuron	Average	RMS
01	tanh	none	0.000644	0.000863
02	sigmoid	none	0.001333	0.001644
03	*sine*	none	0.000486	0.000799
04	sigmoid	none	0.001545	0.001963
05	sigmoid	sigmoid	0.001478	0.001917
06	sigmoid	tanh	0.002036	0.002663
07	tanh	*sine*	0.000656	0.000870
08	tanh	tanh	0.000868	0.001118
09	tanh	sigmoid	0.000520	0.000690
10	*sine*	sigmoid	0.000934	0.001255
11	*sine*	tanh	0.001186	0.001548
12	*sine*	*sine*	0.000934	0.001263

Table 8: Average and RMS error of the training data set for various structures.

on. Here, we only consider the first case. The interval [0, 1] is symmetrically divided in two subintervals.

In Figure 13, the big squares represent the whole input space of the normalized Henon map, while the black small ones are the corresponding input spaces of each submodel. Table 10 gives the average and RMS errors during training, for each of the four submodels resulting from this division of the input space.

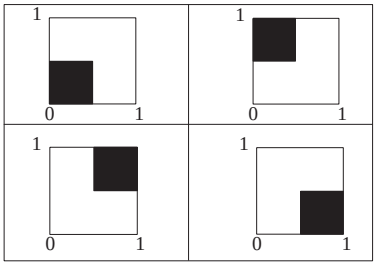

Figure 13: Geometrical representation of input spaces

To operate in a competitive manner, the four networks are driven by an LVQ "controller", trained to recognize the two-dimensional input space, to which each network is trained, and activate the corresponding output. This case corresponds to Figure 1, when n=4 is applied. Figure 14 shows the LVQ network structure. The network has an input layer of two neurons and an

Hidden Layer	tanh	tanh	tanh	sine
Bypass Neuron	none	sigmoid	*sine*	none
Average Error	0.000649	0.000539	0.000660	0.000490
RMS Error	0.000878	0.000539	0.000881	0.000818
$\%Error > 0.1$	0.00%	0.00%	0.00%	0.00%
$\%Error > 0.05$	0.00%	0.00%	0.00%	0.00%
$\%Error > 0.01$	0.00%	0.00%	0.00%	0.00%
$\%Error > 0.005$	0.00%	0.00%	0.00%	0.00%
$\%Error > 0.001$	21.66%	5.86%	16.37%	8.06%
$\%Error > 0.0005$	47.39%	44.66%	53.44%	31.68%

Table 9: Detailed results for the best networks modelling the Henon attractor

	$X_n \in [0, 0.5]$ $X_{n-1} \in [0, 0.5]$	$X_n \in (0.5, 1]$ $X_{n-1} \in (0.5, 1]$	$X_n \in [0, 0.5]$ $X_{n-1} \in (0.5, 1]$	$X_n \in (0.5, 1]$ $X_{n-1} \in [0, 0.5]$
Average Error	0.00015	0.00021	0.0027	0.0003
RMS Error	0.00018	0.00034	0.00038	0.0004

Table 10: Average and RMS error of training data sets

output layer of four, that is, one neuron per class. Two neurons per class are used in the Kohonen layer.

4.6. Comparative Results

The two networks modelling the Henon map are tested on 8000 points of the chaotic orbit, starting from (0, 0). As in case of the logistic map, the average, RMS and distribution of absolute errors are considered. Table 11 and Figures 15a and 15b present the results taken during the testing phase. Again, n=1 means no LVQ controller.

	No LVQ controller	Four output LVQ controller
Average Error	0.00052	0.00023
RMS Error	0.00070	0.00036
$\%Err > 0.005$	0%	0%
$\%Err > 0.001$	5.43%	2.24%
$\%Err > 0.0005$	43.28%	10.42%

Table 11: Comparative results for Henon map models

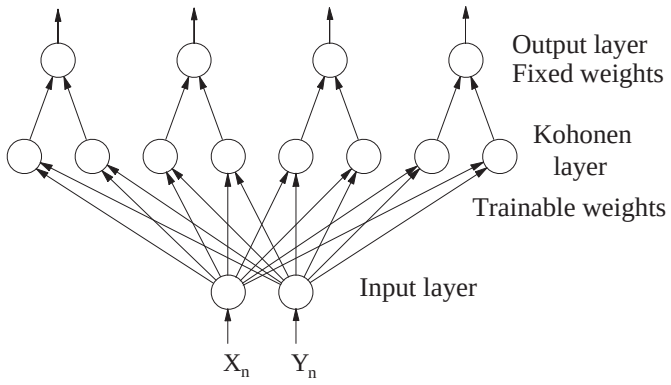

Figure 14: Four Output LVQ network

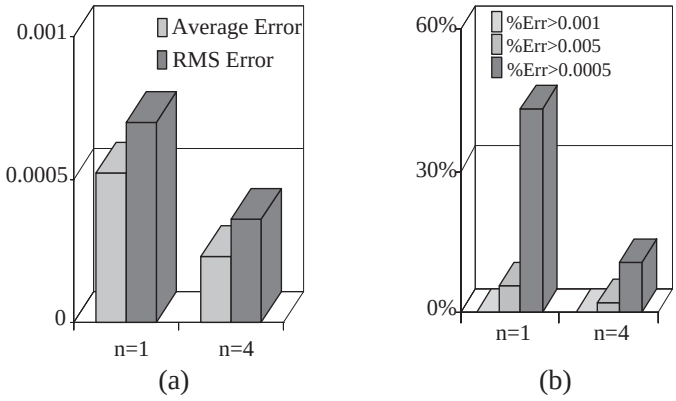

Figure 15: Error of the testing set for single and multiple training: (a) RMS and average error, (b) distribution of absolute error

As shown, the RMS error, when an LVQ controller is used, is almost half the RMS error observed in the single training case. The same applies to the absolute error, when the preselected accuracy level is kept to 0.001. A more impressive picture of how behavior of the model is improved is given by the absolute error, when the "0.0005" error level is considered. In this case, the percentage of data predicted with error higher than 0.0005 is almost divided by four when multiple training is used. That means that the error is less than 0.0005 for about 90% of the active outputs of the network.

5. Chaotic Features During the Recall Phase

5.1. Chaotic Response of the Absolute Error Series

This section examines the response of the absolute error during the recall phase, for the best neural models of the logistic and Henon maps, presented in Sections 3 and 4. For the logistic map, this is the neural model without a bypass unit and λ fixed at 3.93. For the Henon map, the best network is the one with tanh in the hidden and a sigmoidal bypass neuron. These networks were trained using the "single training method", presented in the previous two sections. During the recall phase, a piece of the chaotic orbit, consisting of 8000 points, is used in each case.

Let $X_1, X_2, \ldots, X_n$ be the series of points that constitute the orbit presented to the network and $y_1, y_2, \ldots, y_n$ the series of actual outputs produced. It should be noted that $y_1, y_2, \ldots, y_n$ are the outputs emerging when the corresponding input is error free, meaning that one step prediction is used in any case and no $y_i, i = 1, 2, \ldots, n$, returns as an input value. If $Y_1, Y_2, \ldots, Y_n$ are the corresponding desired outputs, then the term E_i of the absolute error series is expressed by the equation $E_i = |Y_i - y_i|$. To check whether the absolute error behaves chaotically or not, the dominant Lyapunov exponent is calculated for the E_i data series.

The estimation of the Lyapunov exponents is based on Alan Wolf's work [25]. The essential idea of the method is the reconstruction of the underlying attractor, using the so-called embedding scheme. Specifically, an attractor of dimensionality d is reconstructed from a scalar data series through a time delay coordinates method, in a *D=2d+1* embedding space, which mimics the dynamics of the original system [22]. Then, nearby initial condition sets of vectors (in the Euclidean sense of distance) are followed in the new space to reveal stretching properties of the attractor. The dominant Lyapunov exponent is expressed in bits per iteration. The application of the method in data series demands the a priori determination of the following parameters:

- Embedding dimension D: The dimension of the embedding space, the choice of which is based on assumptions made for the dimension of the original attractor.
- Delay time t_d: The reconstructed attractor consists of vectors E which have the form $E = (E_1, E_{1+t_d}, E_{1+2t_d}, \ldots, E_{1+(D-1)t_d})$. From this form of the vector, it is obvious that t_d is the number of intermediate samples skipped during the vector construction process.

- Evolution time t_e: Corresponds to the time (number of samples) that trajectories, starting from nearby initial conditions, are being followed to reveal local stretching of the attractor.

- Maximum distance d_{max}: Corresponds to the maximum permissible distance, in the embedding space, between trajectories starting from nearby initial conditions.

Different values of the above parameters correspond to different aspects of the attractor, thus making it difficult to decide which of the reconstructed spaces mimics the dynamics of the real attractor better. Hence, it is necessary to perform a stability investigation of the estimated exponent.

Results of a complete exploration of the exponents' stability should be represented in terms of a diagram or a table, in a four dimensional space. Considering this impossible, the investigation kept three of the parameters fixed and altered the fourth one in each case. The fixed values of the parameters are called reference values. Several combinations of parameter values have been examined, with the objective of determining an appropriate set of reference values from which the representative Lyapunov exponent emanates.

An additional calculation on the orientation error is also applied in each table, to ensure that no serious divergence of the main stretching direction takes place. Specifically, the percentage of processing steps with orientation error greater than 10 degrees is measured. It should be noted that an orientation error greater than 30 degrees is accepted by the algorithm. In the following, the term acceptable orientation error will be used. This calculation is taken into account for choosing the representative value of the exponent and consequently for determining the set of reference values. The final set, on which stability investigation is based, is the following:

$$(D, t_d, t_e, d_{max}) = (2, 1, 3, 5\%) \tag{2}$$

Although a representative value of the dominant Lyapuvov exponent is used in the following, the main objective is to conclude the existence and stability of a positive exponent.

The four steps of stability investigation are described in the following four diagrams (Figure 16), each one corresponding to one of the parameters being altered.

Tables 12 and 13 summarize the results of the stability investigation. The steps of the stability investigation process are presented in the four rows (starting with D, t_d, t_e, d_{max}). In the first row, the representative value of the expo-

nent (RE) is presented together with the set of reference values of the parameters, on which its calculation is based. The average stability performance of the exponent is given in the last row. The information contained in the columns is the following:

- From - to: The lower and the upper value of the parameter, which changes during a specific step of the stability investigation process.
- Mean Exp.: The average value of the exponent, calculated for every step of the stability investigation process.
- MADR: Mean absolute deviation from the representative value, calculated for every step of the stability investigation process.
- MADA: mean absolute deviation from the mean value of the exponent calculated for every step of the stability investigation process.
- %MADR: Mean absolute deviation from the representative value, presented as a percentage of MADR.
- %MADA: mean absolute deviation from the mean value of the exponent presented as a percentage of MADA.

R.E=0.8842 for $(D, t_d, t_e, d_{max}) = (2, 1, 3, 5\%)$						
	From - to	Mean Exp.	MADR	MADA	%MADR	%MADA
D	$1 \sim 3$	0.88600	0.0842	0.0848	9.52%	9.57%
t_d	$1 \sim 5$	0.91130	0.0709	0.0655	8.02%	7.19%
t_e	$2 \sim 6$	0.80068	0.1111	0.0944	12.57%	11.79%
d_{max}	$5\% - 15\%$	0.80280	0.0813	0.0575	9.19%	7.16%
Average		0.8502	0.0869	0.0756	10%	9%

Table 12: Stability investigation for the exponent of the logistic map

From the fluctuation of the dominant Lyapunov exponent shown in Figure 16 and Tables 12 and 13, it can be concluded that a positive Lyapunov exponent exists for the absolute error series produced when the neural models recall one by one the points of the chaotic orbits, which represent the logistic and the Henon attractors.

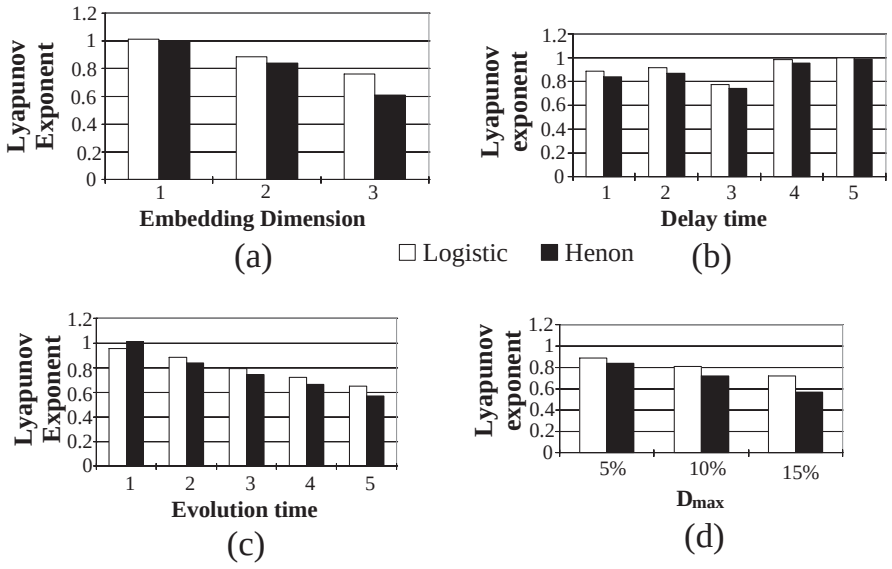

Figure 16: Stability status of the dominant Lyapunov exponent. In each diagram one of the parameters is altered, while the others are kept fixed at their reference values

5.2. Effects on Modelling Chaotic Attractors

To explore the effects of chaotic absolute error on neural system modelling, two significant properties of chaotic systems should be considered.

(1) For a chaotic system $f : A \to A$ the chaotic attractor (limit set of f) is dense to A, meaning that arbitrarily close to any point of A there is an attractor point.

(2) If $f : A \to A$ is chaotic, then f is topologically transitive. That is, for any pair of open sets $U, V \subset A$ there exists $k > 0$ such that $f^k(U) \bigcap V \neq \emptyset$.

Let $f : A \to A$ be a chaotic system and F the simulating neural model. When X_k is presented to the input of the neural network, an output Y_k is produced. If X_{k+1} is the desired output, then $Y_k = X_{k+1} \pm E_k$, where E_k is the corresponding absolute error. The assumption that Y_k belongs to the output space A is reasonable, at least for the models examined in this work. Therefore, due to property (1) of chaotic systems, Y_k can be considered as an attractor point. Then, property (2) leads to the conclusion that Y_k can be produced by

R.E=0.8387 for $(D, t_d, t_e, d_{max}) = (2, 1, 3, 5\%)$						
	From - to	Mean Exp.	MADR	MADA	%MADR	%MADA
D	$1 \sim 3$	0.8147	0.1303	0.1382	15.53%	16.97%
t_d	$1 \sim 5$	0.8767	0.0774	0.0742	9.23%	8.46%
t_e	$2 \sim 6$	0.7646	0.1425	0.1277	16.99%	16.70%
d_{max}	$5\% - 15\%$	0.7078	0.1309	0.093	15.61%	13.14%
Average		0.7910	0.1203	0.1083	14%	14%

Table 13: Stability investigation for the exponent of the Henon map

X_k after a finite number of iterations (again finite accuracy permits considering any number as an open set) leading to $Y_k = X_m, m \neq k+1$. Thus, the neural model of the attractor responds to the input value (point on a specific orbit) producing an orbital point with generally unknown index (unknown position on the orbit). This is a result of the combination of a chaotic input with a chaotic absolute error and can be referred to as output index shifting. This term expresses the property of chaotic ordering of orbital points in the output space. Since any output value is an attractor point, an output set reliably representing the geometrical features of the attractor can be produced. Specifically, if an initial input value X_1 is applied to the network, the attractor point X_k $(k \neq 2)$ will be produced, which, returning as an input value will create an output equal to the attractor point X_m $(m \neq k+1)$ and so on. In conclusion, the neural model of a chaotic attractor is able to reliably reconstruct the attractor as a set of points with specific geometry in the attractor space.

Although attractor reconstruction seems to be possible, the neural models theoretically fail to effectively capture the evolution of the system, since $G = (X_1, X_k, X_m, \ldots, X_p)$ do not represent the desired orbital sequence $g = (X_1, X_2, X_3, \ldots, X_n)$. From a more practical point of view, a question that might arise is whether the two sequences, G and g (corresponding to F and f functions), are close enough, so that $X_1, X_k, X_m, \ldots, X_p$ could be considered as a satisfying approximation of $X_1, X_2, X_3, \ldots, X_n$. In other words, it is implied that a part of each error value acts as a dithering noise, improving the approximating properties of the system.

To investigate this possibility, the first two steps of data presentation are considered. When X_1 is presented, the network produces $F(X_1) = X_2 \pm E_1$, where X_2 is the desired output and E_1 is the corresponding absolute error. The original system evolves, fed with X_2 as an input value for the next step, while the neural model is fed with $X_2 \pm E_1$. The values X_2, $X_2 \pm E_1$ could be considered as nearby initial conditions (E_i is small for good neural models), if f and F

were identical. Although they are not, the terms "nearby initial conditions and exponential divergence" will be used. Since F simulates f, a calculation of their exponential divergence forms a criterion for the level of accuracy of the simulation of the orbital sequence, when compared with the corresponding original exponential divergence caused in f. During the second step, $X_2 \pm E_1$ produces $F(X_2 \pm E_1) = f(X_k) \pm Ek$, implying that $X_2 \pm E_1 = X_k$ as previously depicted. Now $f(X_k)$ can be expressed as $X_3 \pm d3$, where $|d_i| \leq R$ (R being the range of system values). Thus, $F(X_2 \pm E_1) = X_3 \pm d_3 \pm E_k$. The term d_i represents the exponential divergence due to sensitivity to initial conditions, while E_j is a member of the produced absolute error series during the one step prediction process. In each step of evolution a d_i and an E_j term are present.

g: sequence produced by $f(X_i)$	G: sequence produced by $F(X_i)$
X_2	$X_2 \pm E_1$
X_3	$X_3 + d_3 \pm E_k$
X_4	$X_4 + d_4 \pm E_m$
$\vdots$	$\vdots$
X_n	$X_n + d_n \pm E_q$

Table 14: Original and simulated chaotic orbits

Successful neural models require low E_j values, presuming that the original exponential divergence (observed in f) maybe delayed or speeded up unpredictably, due to the chaotic form of E_j, but not eliminated. Numerical tests indicate that the original orbit and the approximating sequence are locally submitted to exponential divergence. Figure 17, shows a typical picture of local behavior.

What can be concluded is that the output sequence owns a restricted approximation ability, the depth of which seems to obey local features, emanating from the chaotic properties of the added error, produced by neural models in a step by step presentation of a chaotic orbit during the recall phase. The neural models proposed in this paper are able to produce a reliable geometrical representation of the simulated attractor. However, their ability of simulating the evolving system is limited. This limitation cannot be directly deduced when the RMS error is considered. The final criteria should also take into consideration the distribution of the absolute error and the behavior of the absolute error series.

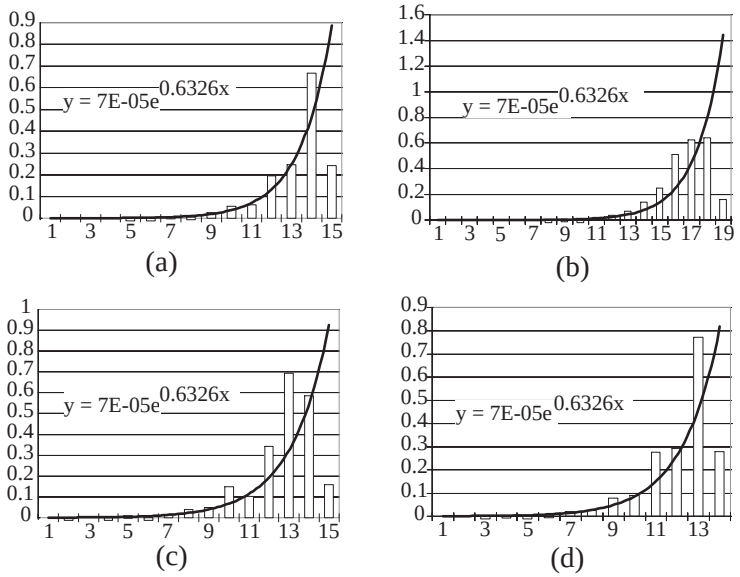

Figure 17: Exponential divergence in local behavior: (a) between F and f starting from 0.1 (b) if f between trajectories starting from f(0.1), f(0.1)+E_1 (c) between F and f starting from 0.3 (d) if f between trajectories starting from f(0.3), f(0.3)+E_1

6. Chaotic Properties of Simple Neural Networks During the Training Phase

6.1. Overview

This section examines the chaotic response of small sized Back Propagation neural networks. Two cases are considered. First, a simple autoassociator is trained using the Back Propagation algorithm for a pre-determined number of epochs. The algorithm does not converge, meaning that the network does not learn the patterns well by the end of the training phase. To indicate the chaotic properties of the network, additional training is applied and two kinds of data series are created: one concerning the weight vector and the other the input patterns producing outputs with the minimum absolute error after each training epoch. The dominant Lyapunov exponent and its stability properties are investigated for each data series.

Subsequently, the simple autoassociator examined in the first case is im-

proved and the desirable convergence is achieved during a certain number of epochs. Additional training is again applied to the new network, and a new data series is created, consisting of input patterns producing outputs with the minimum absolute error after each training epoch. This new data series is submitted to a similar Lyapunov exponent investigation as the previous ones, to demonstrate chaotic response. However, its chaotic properties do not permit direct conclusions for the response of the data series consisting of the outputs that correspond to the best learned inputs, namely the best estimations. Topological conjugacy and numerical verification are the main tools for the exploration of the behavior of the best estimation series. Finally, series of RMS errors consisting of values calculated after each epoch of additional training are examined. Investigation of the RMS error is performed in correspondence with the distribution of the absolute error of the data set.

6.2. Non Learning Networks

6.2.1. Network Description

The first step in the current investigation of chaotic properties of neural networks is based on a simple network that does not meet desirable convergence during the learning phase. What leads to the examination of a non-converging network, is the initial consideration that it is more probable to discover chaotic properties in an oscillating system than in one, which is typically stable within predefined accuracy limits.

The neural network in discussion is a simple autoassosiator consisting of only three units, one each for the input, hidden, and output layers. The input and output units apply the linear transfer function, while the hidden one is a sigmoidal node as can be seen in Figure 18. Table 15 gives the parameters of the training phase.

The network fails to learn after 10,000 epochs retaining an average error of 0.1784 and an RMS error equal to 0.2066.

Since these are the best results achieved, it is considered that the behavior of the network is the same, regardless of the length the training phase. Thus, additional training of 120 epochs is applied to the network, representing its behavior. Series of data are produced for the weights W_1, W_2 and for the best learned input patterns. Specifically, after each epoch of training, the new values of weights are added to the data series. In addition, the network is submitted to a test on the data set and the input pattern corresponding to the minimum absolute output error is found and added to the series of the best learned

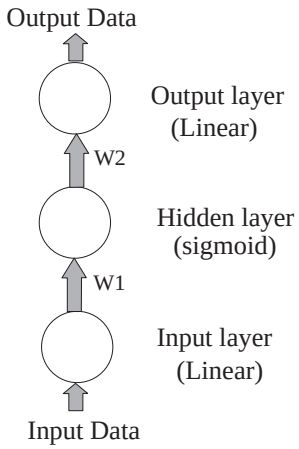

Figure 18: Non converging autoassociator

input patterns. In the following, the dominant Lyapunov exponent analysis is described for these three data series, to investigate the chaotic properties of the network during the learning phase.

6.2.2. The Framework of the Estimation of the Dominant Lyapunov Exponents

The estimation of the dominant Lyapunov exponents is performed in a similar way as in Section 5. However, the small size of the data series and the availability of two measurements in some cases, pose the demand of a detailed presentation of the calculation process of the exponents. Thus, the summarizing tables of Section 5 are abandoned and detailed tables and diagrams are adopted. The specific criteria used for the selection of the representative exponent value are described for each case.

6.2.3. Weight Series Investigation

Here, the two weight series produced during additional learning are subjected to a Lyapunov exponent estimation procedure. A question that might arise is whether the amount of data used is enough for estimating the dominant Lyapunov exponent. According to Allan Wolf [25], the amount of data needed for the stabilization of the exponent lies between 10^d and 30^d, where d is the dimension of the investigated attractor. For $d = 2$, as in this case, the minimum

Type of Network	Back Propagation autoassosiator with one input, one hidden, and one output unit
Learning Rate	0.1
Momentum	0.85
Range of initial weight space	[-0.1, 0.1]
Data set used for training	500 pairs uniformly distributed
Epochs	1
Pattern presentation	Random
Max Number of iterations	10000

Table 15: Parameters of training

amount of data needed is 100 values. Indeed, each of the two data series consists of 120 data values, which is just above the lower limit. This fact does not pose a problem, since Wolf's law concerns cases in which only one measurement is available. For this study, measurements of both W_1 and W_2 have been taken. Figures 19a and 19b show the evolution of the Lyapunov exponent during its estimation, for the W_1 and W_2 data series.

The range of the attractor is 0.299 on W_1 axis and 0.021 on the W_2 axis, as observed from the weight data series produced during additional training.

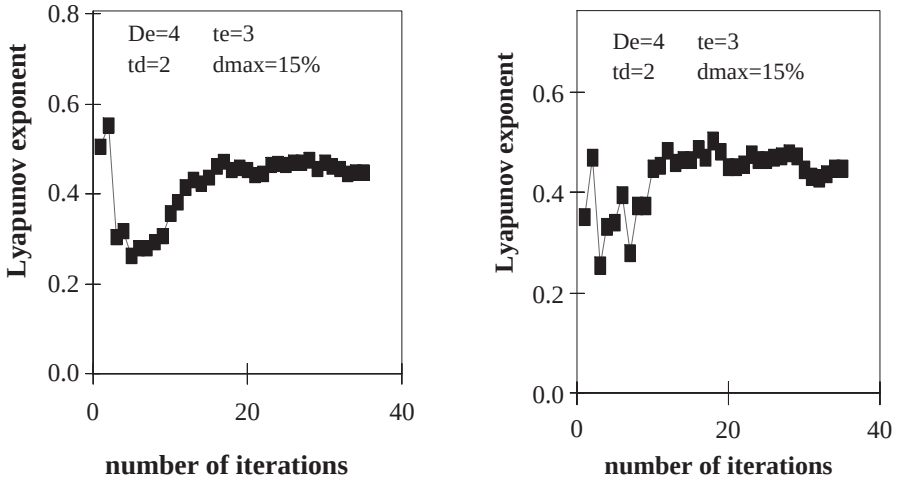

Figure 19: (a) Evolution of the Lyapunov exponent for the W_1 series (b) Evolution of the Lyapunov exponent for the W_2 series

Delay time (t_d) selection: The choice of delay time is critical because it determines how scalar data should be combined to form the vectors that reconstruct the embedding space. Table 16 and Figure 20a present the stability status of the exponent, when t_d varies from 1 to 5. The reference values for the rest of the parameters are $D_e = 4$, $t_e = 3$, and $d_{max} = 15\%$. These values along with $t_d = 2$ are used as a set of reference values since they satisfy the demand of internal reliability under the following criteria:

(1) The fact that the attractor is reconstructed twice, using two measurements (for both W_1 and W_2) is helpful, leading to the consideration that the exact exponent value ought to be somewhere in the interval defined by the closest values. More clearly, if $L(W_n^1)$ and $L(W_n^2)$ represent data in the second and the third columns of Table 6.2, then the Lyapunov exponent lies in the interval $[L(W_k^1), L(W_k^2)]$, with $L(W_k^1)$ and $L(W_k^2)$ satisfying the equation:

$$|L(W_k^1) - L(W_k^2)| = min|L(W_n^1) - L(W_n^2)|, \quad n = 1, 2, 3, 4, 5. \tag{3}$$

(2) The orientation error should take acceptable values.

(3) The estimated value ought to be as close as possible to the mean value of all the values calculated.

Criterion (1) is the dominant one while (2) and (3) play an advising or confirmation role. These requirements are satisfied by the values corresponding to $t_d = 2$, meaning that the representative Lyapunov exponent lies in the interval $[0.441, 0.453]$.

t_d	Lyapunov Exponent		Orientation Error	
	W_1	W_2	W_1	W_2
1	0.468	0.695	23.6%	5.4%
2	0.441	0.453	5.7%	8.3%
3	0.209	0.218	2.8%	0.0%
4	0.383	0.407	2.9%	2.9%
5	0.481	0.463	6.0%	3.0%

Table 16: Stability investigation with respect to t_d

Embedding Dimension (D_e) selection: As already mentioned, the embedding dimension is given by the equation *D=2d+1*, where d is the dimension of the underlying attractor. Although this seems to be the right choice, problems might arise in finding replacement points or in satisfying orientation constraints [25]. It has been determined that *D=2d* is sufficient [12]. Table 17 and Figure 20b show the stability status of the exponent, as D_e takes the values 4 and 5.

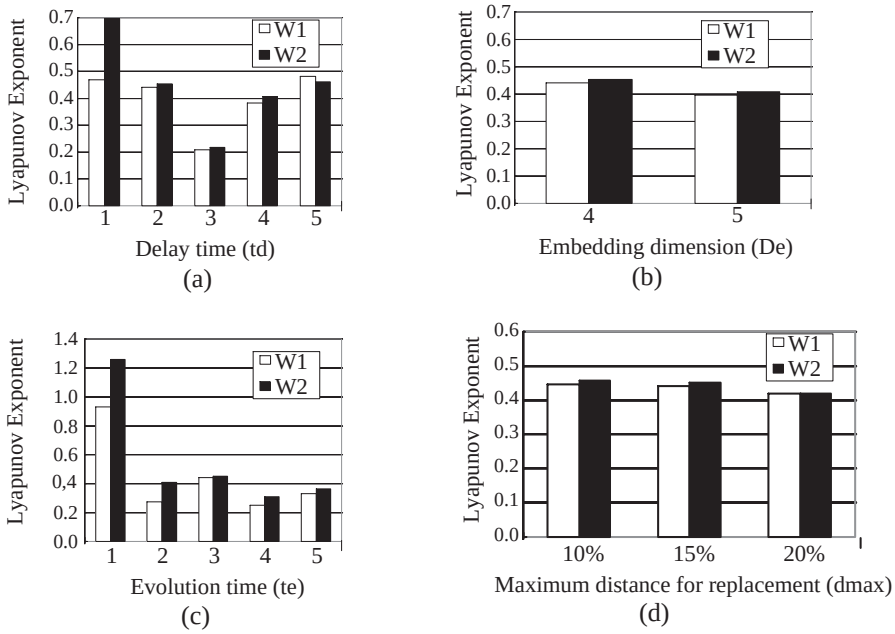

Figure 20: Stability with respect to: (a) (t_d), (b) D_e, (c) t_e, and (d) d_{max}

D_e	Lyapunov Exponent		Orientation Error	
	W_1	W_2	W_1	W_2
4	0.441	0.453	5.7%	8.3%
5	0.395	0.408	2.9%	0.0%

Table 17: Stability investigation with respect to D_e

The requirement of the most adjacent values is satisfied by the choice $D_e = 4$ (corresponding to $D_e = 2d$). Since orientation errors are acceptable and the estimated exponents are close to the mean value, the representative Lyapunov exponent again lies in the interval $[0.441, 0.453]$.

Evolution time (t_e) selection: The proper choice of evolution time is essential, for it leads to the calculation of local Lyapunov exponents. The overall procedure of Lyapunov exponent estimation is fragmented in steps determined by the evolution time, and local stretching of the present piece of the attractor is calculated in each step. Lyapunov exponents are calculated for various values of t_e as shown in Table 18 and presented in stability diagram in Figure 20c.

As can be seen, the criteria for choosing a value to represent the exponent

t_e	Lyapunov Exponent		Orientation Error	
	W_1	W_2	W_1	W_2
1	0.931	1.257	8.1%	8.1%
2	0.272	0.408	5.5%	1.8%
3	0.441	0.453	5.7%	8.3%
4	0.252	0.309	7.4%	3.8%
5	0.330	0.367	9.5%	4.8%

Table 18: Stability investigation with respect to t_e

are satisfied by the value $t_e = 3$. Thus, the representative Lyapunov exponent is chosen to be in the interval $[0.441, 0.453]$.

Selection of maximum distance for replacement ($d_{\max}$): Trajectories starting from nearby initial points in the embedding space are followed for time (iterations) that is a multiple of t_e, and local Lyapunov exponents are calculated. A choice of new initial points, (specifically a new initial point near fidusial trajectory refered in [25]) is made when a pre-chosen maximum distance between trajectories is reached. The stability of the exponent with respect to d_{max} is presented in Table 19 and Figure 20d.

d_{max}	Lyapunov Exponent		Orientation Error	
	W_1	W_2	W_1	W_2
10%	0.446	0.457	0.0%	5.6%
15%	0.441	0.453	5.7%	8.3%
20%	0.420	0.419	22.2%	5.6%

Table 19: Stability investigation with respect to d_{max}

The estimated values are all very close together as shown in this table. The value $d_{max} = 20\%$ is rejected because of the relatively large value of the orientation error for W_1. Values of the exponent corresponding to $d_{max} = 10\%$, satisfy criterion (1), but values corresponding to $d_{max} = 15\%$, (which have been chosen for all the other cases above) are closer to the mean value. The best approach in this case is to consider that the Lyapunov exponent lies in the intersection of the intervals defined by the couples of the estimated exponents in Table 19. This final estimation is the best, since it improves the estimations of all other cases, by decreasing the interval in which Lyapunov exponent was supposed to lie.

6.2.4. Best Learned Input Pattern Series Analysis

Since the weight update takes place after the pattern presentation, and patterns are randomly presented to the network during the learning phase, there ought to be some kind of chaotic pattern competition emerging from the chaotic response of the weight vector. To check for this hypothesis, the best-learned input pattern is kept after each learning epoch by testing the network on the data set. The series of data selected corresponds to the input pattern that exhibits the minimum absolute output error in each step of learning process. From another point of view, the produced data series corresponds to the itinerary, within the limits of the input space, of the best learned input pattern during training. The chaotic properties of this itinerary are realized using the same methodology as in the previous section, and the stability tables and diagrams of the estimated dominant Lyapunov exponent are presented below. The set of parameter values found to satisfy internal reliability are $t_d = 1$, $D_e = 2$, $t_e = 2$, and $d_{max} = 15\%$. The size of the attractor is 0.338, as calculated from the produced data series.

Delay time (t_d) selection: Table 20 and Figure 21a give a complete picture of how Lyapunov exponent fluctuates when the embedding space is reconstructed from different vectors produced from series of data.

t_d	Lyapunov Exponent	Orientation Error
1	0.916	17.5%
2	0.680	21.1%
3	1.089	29.8%
4	0.990	24.6%
5	0.879	17.5%

Table 20: Stability investigation with respect to t_d

As can be seen, the orientation error is larger than the corresponding one in the previous section, but it still takes acceptable values. To choose the representing value of the exponent from Table 20 two criteria are considered:

(1) The value chosen should be as close as possible to the average of the estimated exponents.

(2) The orientation error ought to take an acceptable value.

The value 0.916, corresponding to $t_d = 1$, satisfies the criteria, and is chosen to represent the dominant Lyapunov exponent in this case.

Embedding Dimension (D_e) selection: Since the attractor of the best learned pattern is of dimension $d \leq 1$ for this network, three cases are examined, corresponding to the projection of the data series to an embedding space with

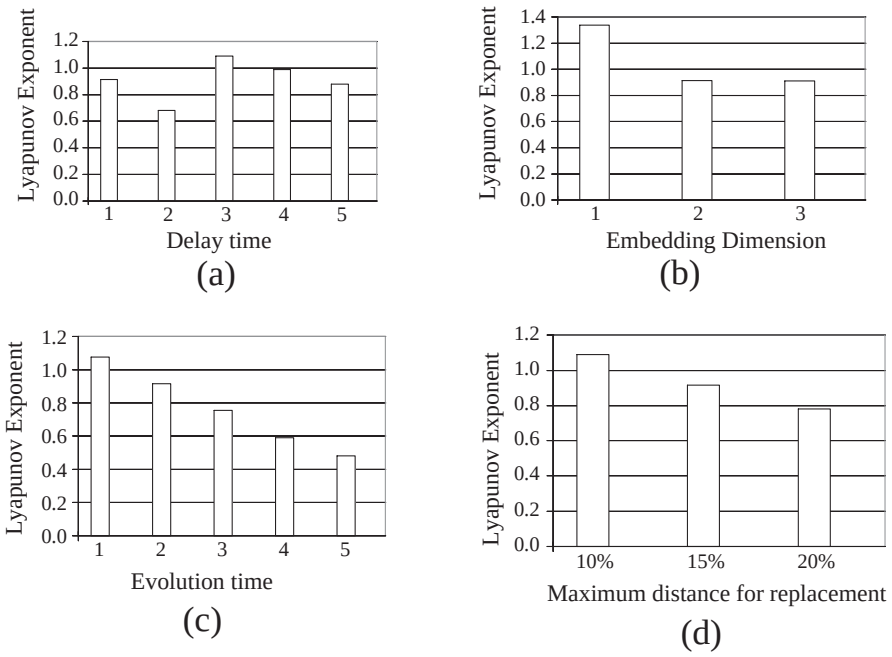

Figure 21: Stability with respect to: (a) t_d, (b) D_e, (c) t_e, and (d) d_{max}

dimension D_e=1, 2 and 3. Table 21 and Figure 21b present the estimated values of the exponent.

D_e	Lyapunov Exponent	Orientation Error
1	1.337	0.0%
2	0.916	17.5%
3	0.910	21.1%

Table 21: Stability investigation with respect to D_e

The values corresponding to $D_e = 2$, $D_e = 3$ are close together and could both be considered representative, while $D_e = 1$ is rejected for it seems to overestimate the exponent. The choice is made for $D_e = 2$ for it is closer to the average and retains a lower orientation error.

Evolution time (t_e) selection: Results on exponent stability with respect to the evolution time are presented in Table 22 and Figure 21c. The choice of the representative Lyapunov exponent is again 0.916 corresponding to $t_e = 2$.

This may seem to contradict the first of the two criteria, since the value

t_e	Lyapunov Exponent	Orientation Error
1	1.076	19.0%
2	0.916	17.5%
3	0.755	28.9%
4	0.590	17.9%
5	0.480	27.3%

Table 22: Stability investigation with respect to t_e

0.755, corresponding to $t_e = 3$, is closer to the average, but there are some reasons for rejecting this case. First, the orientation error is much larger than for the selected value. In addition, if $t_e = 3$ is selected, the whole study has to start from the beginning in order to specify a new set of reference values. The stability status emerging from these new reference values is no more satisfactory. Specifically, the new reference values ($t_d = 4$, $D_e = 2$, $t_e = 3$, and $d_{max} = 15\%$) exhibit larger orientation error (24.3%), while an underestimation of the dominant Lyapunov exponent is possible, because when t_e increases, folding properties of the attractor may be captured. This underestimation is confirmed by the exponent value calculated when this new set of parameters is applied to the data series (0.708). Finally, stability investigation based on the new set of parameters, supports the selection of $t_e = 2$, since this value better satisfies the criteria. Hence, there seems to be a kind of cyclic reference between $t_e = 2$ and $t_e = 3$. The final selection (set of reference values containing $t_e = 2$) is based mainly on the computations of the orientation error.

Selection of the maximum distance for replacement (d_{max}): Maximum distance for replacement is the last parameter checked in order to conclude the Lyapunov exponent analysis. As shown in Table 23 and Figure 21d, the choice $d_{max} = 15\%$ satisfies the criteria. This final investigation confirms internal reliability of the set of parameters selected for the estimation of the representative dominant Lyapunov exponent. Thus, the value 0.916 bits/iteration is considered a good estimation expressing the average rate of the stretching properties of the best-learned pattern attractor.

d_{max}	Lyapunov Exponent	Orientation Error
10%	1.089	35.1%
15%	0.916	17.5%
20%	0.781	22.8%

Table 23: Stability investigation with respect to d_{max}

It can be concluded that random presentation of the patterns during the learning phase leads to a chaotic updating of the weights, which in turn leads to the chaotic movement of the best learned input pattern.

6.3. Pattern Competition in Converging Networks

When random presentation of patterns in the learning set takes place, pattern competition is an obvious learning phase property of neural networks, until convergence is reached. This is depicted in the following set of diagrams (Figure 22) which concern a Back Propagation neural network with functional link inputs trained on the chaotic attractor of the logistic map for $\lambda = 3.93$ (Section 3). The final RMS error achieved is 0.00051 after 30,000 iterations (3,000 epochs).

The question that arises is whether pattern competition is present after desirable convergence is reached. Even more, whether this competition, given its existence, exhibits chaotic properties and how this may affect the network's behavior.

Therefore, the chaotic behavior of a network during the learning phase should be examined, to determine if it is independent from what desirable convergence is considered to be in each case. It should be noted that, since convergence is reached, the size of the attractor is expected to be small. Also, progressive increase of the required accuracy acts as a guideline to the state of the practically unobservable attractors, especially in applications such as chaos control with neural models or chaotic systems' neural modelling.

For the purpose of chaotic analysis, a much simpler network is considered, than the one of Figure 22. In fact, an improvement of the non-converging autoassosiator in Section 6.2 was selected for Lyapunov exponent analysis. Various improvements of this network are presented in Table 24. The columns of the table contain results for the networks trained to different training intervals. Compared to the network in Section 6.2, the selected network (column in bold characters) has an additional node in the hidden layer, which is connected forward and backwards. The network was trained to the interval [0, 0.05] using the training parameters in Table 15.

6.3.1. Lyapunov Exponent Estimation for the Best Learned Input

The results of Lyapunov exponent estimation are presented for the data series of the best learned input patterns, produced during an additional training of 120 epochs. It should be noted that the best-learned input pattern is not unique. Figure 23 shows the distribution of the absolute error with respect to the input

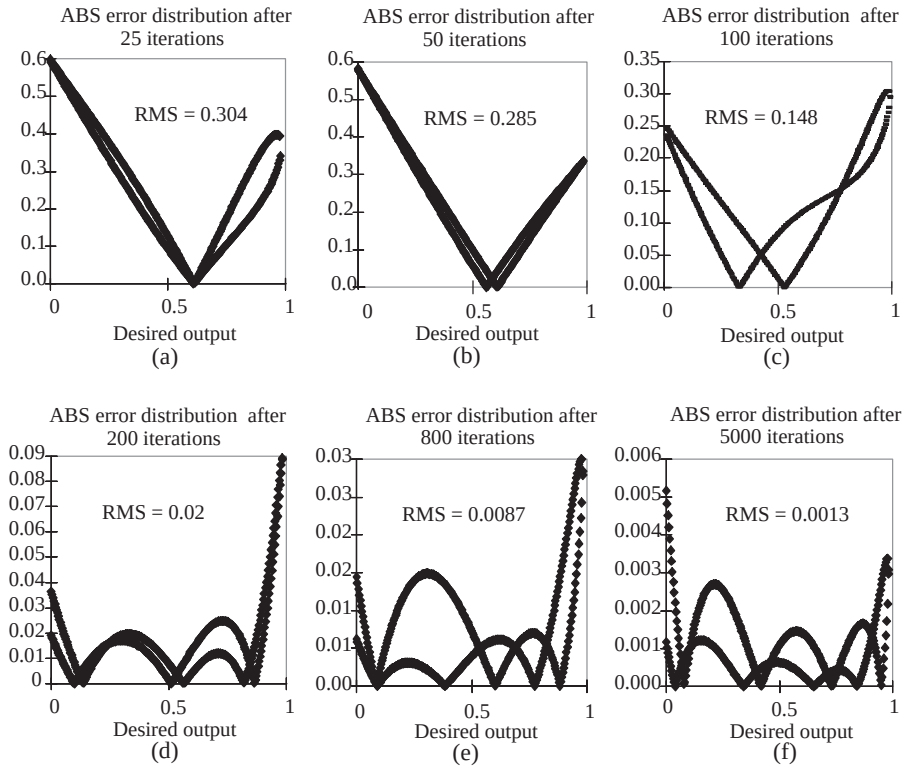

Figure 22: Pattern competition during learning phase before convergence is reached

patterns after the n_{th} epoch. As can be seen, two values of the data set are best learned after each training epoch, implying a two dimensional attractor embedded to a four or five dimensional space.

Two measurements were taken by applying the method of Section 6.2. The set of reference values for the parameters of exponent estimation are $t_d = 4$, $D_e = 4$, $t_e = 3$, and $d_{max} = 15\%$. The range of attractor emerging from data series is 0.0084 for MIN_1 axis and 0.0072 for MIN_2 axis. The term MIN is used as an association to the fact that minimum absolute error is a requirement for the corresponding outputs.

Delay time (t_d) selection: Table 25 and Figure 24a present the results of stability investigation when time delay is altered.

The value $t_d = 4$ results to the best estimation of the exponent in terms of the criteria (1), (2) and (3) set in Section 6.2, for the case in which two

Network with one unit in the hidden layer					
Training interval	1.0000	0.5000	0.2500	0.1000	0.0500
Average Error	0.1784	0.0940	0.0492	0.0183	0.0092
RMS Error	0.2066	0.1093	0.0590	0.0211	0.0106
Network with two units in the hidden layer					
Training interval	1.0000	0.5000	0.2500	0.1000	0.0500
Average Error	0.0102	0.0032	0.0015	0.0007	0.0004
RMS Error	0.0115	0.0039	0.0019	0.0008	0.0004

Table 24: Stability investigation with respect to t_e

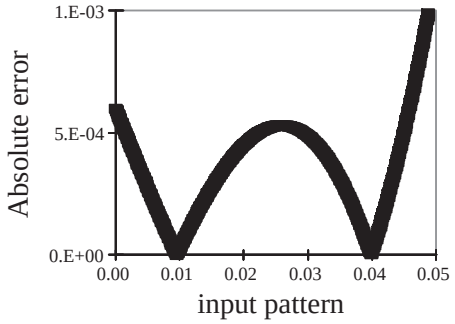

Figure 23: Distribution of absolute error for the improved network

measurements are available.

Embedding Dimension Selection: Table 26 and Figure 24b present the results of stability investigation when embedding dimension is altered.

The parameter value $D_e = 4$ is chosen, and again the Lyapunov exponent lies in $[0.328, 0.329]$.

Evolution time selection: Table 27 and Figure 24c present the results of stability investigation with respect to the evolution time t_e.

The choice for the representative Lyapunov exponent is in $[0.328, 0.329]$, corresponding to $t_e = 3$.

Selection of maximum distance for replacement (d_{max}): Table 28 and Figure 24d present the results of stability investigation with respect to the maximum distance, d_{max}.

As can be seen in Table 28, the application of the three different parameter values to the data series results in equal estimations. Acceptable orientation error when $d_{max} = 15\%$ makes this choice acceptable. Thus, the set of parameter values seem to be internally reliable, presuming that the number in the

t_d	Lyapunov Exponent		Orientation Error	
	MIN_1	MIN_2	MIN_1	MIN_2
1	0.305	0.269	2.7%	8.1%
2	0.331	0.305	2.8%	5.6%
3	0.180	0.209	5.7%	2.9%
4	0.329	0.328	2.9%	0.0%
5	0.419	0.382	0.0%	3.0%

Table 25: Stability investigation with respect to t_d

D_e	Lyapunov Exponent		Orientation Error	
	MIN_1	MIN_2	MIN_1	MIN_2
4	0.329	0.328	2.9%	0.0%
5	0.327	0.310	0.0%	0.0%

Table 26: Stability investigation with respect to D_e

interval $[0.328, 0.329]$ is a reasonable representation of the dominant Lyapunov exponent governing the average stretching properties of the attractor.

6.3.2. Chaotic Response of the Best Estimation

As described in Section 6.3.1, the best-learned input pattern data series exhibit a positive Lyapunov exponent. Although it may seem obvious, one cannot directly conclude that data series produced for the best estimations in output respond chaotically as well. The fact that output may not follow input's behavior is supported by the absolute error series produced for the best-learned patterns. The absolute error is the absolute difference between the best input minus the best output after each training epoch for the case of the autoassosiative network The absolute error data series exhibit a positive dominant Lyapunov exponent (Lcoef) as shown in Figures 25a and 25b. This means that there is a possibility for the error to act as a dithering noise, normalizing or even applying periodic properties to the produced output.

To examine the output's behavior the property of topological conjugacy should be taken in to consideration. Two maps $f : A \to A$ and $g : B \to B$, are topologically conjugate if there exists an homeomorphic map $h : A \to B$, which establishes a direct correspondance between the two maps as can be seen in Figure 26.

Definition. Two maps $f : A \to A$, and $g : B \to B$, are said to be

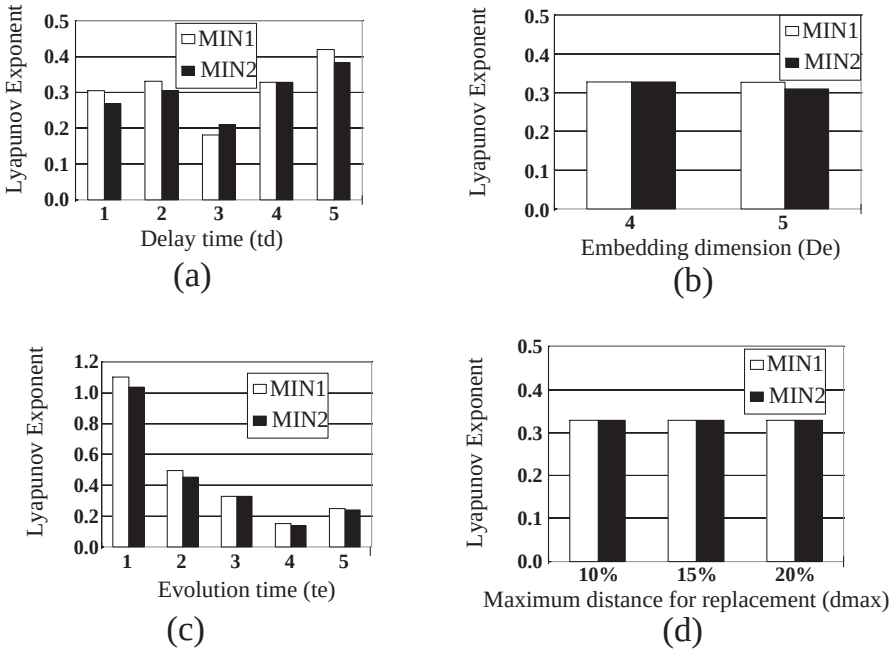

Figure 24: Stability with respect to (a) t_d, (b) D_e, (c) t_e, and (d) d_{max}

topologically conjugate via an homeomorphism $h : A \to B$, if for any $x \in A$ the equation $h(f(x)) = g(h(x))$ stands [6].

Since the sigmoid and linear functions are homeomorphic, a single node is first considered and topological conjugacy is examined for the best learned input and output pattern series.

Let $f : A \to A$, $f(x_n) = x_{n+1}$, A being the best learned input space, and f the underlying map determining the best learned input pattern, x_{n+1}, after the $(n+1)_{th}$ training epoch, as a function of x_n.

Let $g : B \to B$, $g(y_n^i) = y_{n+1}^i$, B being the space of the best estimations, and g the underlying map determining the best estimation, y_{n+1}^i, after the $(n+1)_{th}$ training epoch, as a function of y_n^i (in fact "i" denotes the local character of y as can be seen later in this section).

Let $h(w, x) : A \to B$, be the transfer function of the node, either sigmoid or linear, w being the weight vector leading to the node.

Then, the following equations hold:

$$h(f(x_n) = h(x_{n+1}) = y_{n+1}^i \tag{4}$$

t_e	Lyapunov Exponent		Orientation Error	
	MIN_1	MIN_2	MIN_1	MIN_2
1	1.102	1.037	0.0%	0.0%
2	0.495	0.453	3.8%	0.0%
3	0.329	0.328	2.9%	0.0%
4	0.152	0.138	0.0%	0.0%
5	0.247	0.240	0.0%	0.0%

Table 27: Stability investigation with respect to t_e

d_{max}	Lyapunov Exponent		Orientation Error	
	MIN_1	MIN_2	MIN_1	MIN_2
10%	0.329	0.328	0.0%	0.0%
15%	0.329	0.328	2.9%	0.0%
20%	0.329	0.328	8.8%	0.0%

Table 28: Stability investigation with respect to d_{max}

$$g(h(w, x_n)) = g(y_n) = y^i_{n+1} \tag{5}$$

In (4), f produces the best learned input pattern x_{n+1} while $h(w, x)$ projects it to the output space. It is obvious that this projection meets the best-estimated output, since they are coupled in the process of their determination. Similarly, $h(w, x)$ projects all best learned input values to the output space, while g in (5) produces the next best estimated output. Thus, both the left-hand members of the above equations produce the best estimation and $f(x)$ and $g(y_i)$ are topologically conjugate. The equivalence of $f(x)$ and $g(y_i)$, in terms of their dynamic behavior directly leads to the conclusion that the best estimation in output moves chaotically following the corresponding input's behavior for a single node network.

For the converging autoassociator, what has been proven so far is that the best input pattern map is topologically conjugate to the local maps corresponding to the best local estimations y^1 and y^2 of the hidden nodes. Considering that the output node is fed with their linear combination $I = y^1 w_3 + y^2 w_4$, topological conjugacy of this sum to the best input pattern map should be proven. This can be achieved via the homeomorphic function $H(w, x) = h_1(w_1, x)w_3 + h_2(w_2, x)w_4$, where (w_1, w_2) is the weight vector leading to the hidden layer and (w_3, w_4) the weight vector leading to the output node. More clearly:

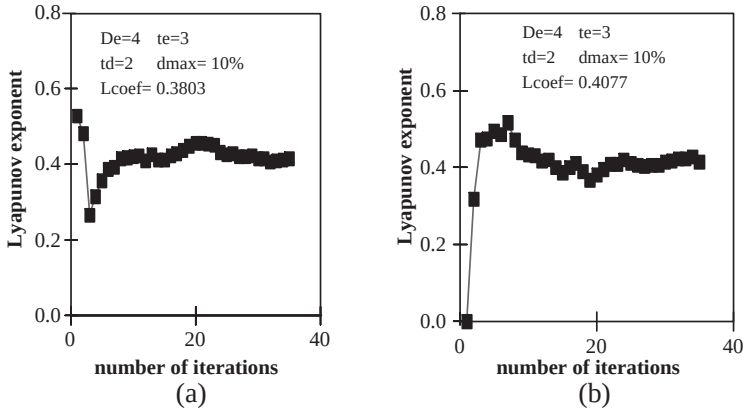

Figure 25: Evolution of the Lyapunov exponent for absolute error series corresponding to (a) MIN_1 data series and (b) MIN_2 data series

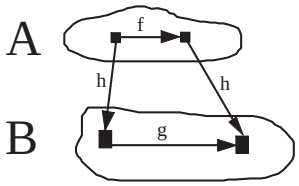

Figure 26: Schematic diagram of topological conjugacy between f and g

- Let $f : A \to A, f(x_n) = x_{n+1}$.
- Let $G : I \to I, G(I_n) = I_{n+1}$, I being the input space for the output node, and G the underlying map determining the best local estimation, I_{n+1}, feeding the output node after the $(n+1)$ training epoch.
- Let $H : A \to I, H(w, x) = h_1(w_1, x)w_3 + h_2(w_2, x)w_4$.

Then:

- $H(f(x_n)) = h_1(w_1, f(x_n))w_3 + h_2(w_2, f(x_n))w_4 = h_1(w_1, x_{n+1})w_3 + h_2(w_2, x_{n+1})w_4 = y^1_{n+1}w_3 + y^2_{n+1}w_4 = I_{n+1}$

and

- $G(H(x)) = G(h_1(w_1, x_n)w_3 + h_2(w_2, x_n)w_4) = G(y^1_n w_3 + y^2_n w_4) = G(I_n) = I_{n+1}$.

Hence, topological conjugacy between the best input pattern map and the best local estimation corresponding to the input of the output node holds. Considering what has been proved before for a sigmoidal or linear node it follows that the input in of the output node is topologically conjugate to its output y_n. To determine whether the best estimation in the output and best learned input pattern series are topologicaly conjugate, the property of transitivity of topological conjugacy should be examined.

Let p be conjugate to q via u_1 and q conjugate to r via u_2. Then, $u_1(p) = q(u_1)$ and $u_2(q) = r(u_2)$. Applying u_2 to the first of the two equations as a composition term, it is evident that $u_2(u_1(p)) = u_2((q(u_1)) = r(u_2(u_1))$.

Thus, p is topologically conjugate to r via the composition of u_2 and u_1, and the best input pattern and best output pattern are conjugate via H, since the output node of the autoassosiative network under discussion is linear.

The best local estimations are chosen with regard to minimum output absolute error, in order to retain the correspondence of input best learned patterns and output best estimations independent of the itinerary used. That means that direct correspondence of input - output patterns is the same as in the case in which local estimations are used as intermediate process stations. The best local estimations, in the way determined, are of no special significance. They are just used as a tool to establish input - output topological conjugacy. The above considerations for the best estimated output series are numerically verified as shown in Figures 27a and 27b, where the final values representing the dominant Lyapunov exponent are close to the values determined for the best learned input pattern attractor.

The application of the method of topological conjugacy to the minimum absolute error series was avoided and only numerical tests were presented. To prove topological conjugacy between minimum absolute error and best input pattern series, an underlying map $E : E \to E, E(e_n) = e_{n+1}$ should be considered, applying the minimum error at step $(n + 1)$ as a function of minimum error at step n. The function via which topological conjugacy may be formulated is given by the equation: $h_e : A \to Eh_e(x) = |F(x) - Y(x)|$, $F(x)$ being the transfer function of the network, and $Y(x)$ the underlying function applying the desired output for a given input value. The fact that $Y(x)$ is in general unknown, does not permit any conclusions on whether $h_e(x)$ is homeomprphic or not, thus making the method useless in this case.

The same applies to the series produced for the desired output corresponding to the best-learned input. In that case, $f : A \to A, f(x_n) = x_{n+1}$ is considered for the best learned input pattern and $d(y_n) = y_{n+1}$ for best learned output pattern. The function $Y(x)$ (Y being the underlying function producing

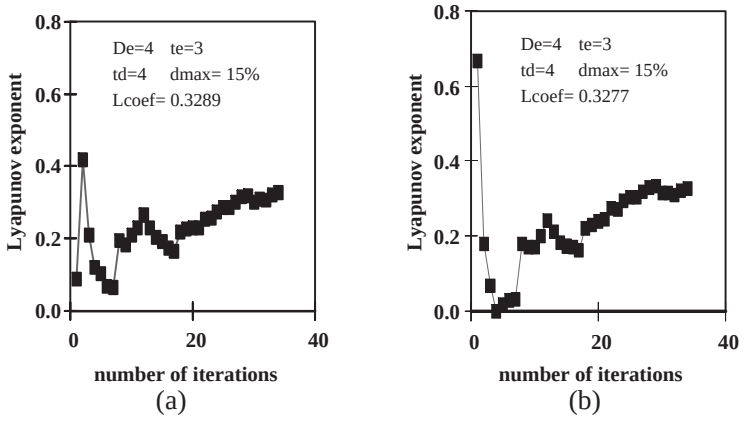

Figure 27: Evolution of the Lyapunov exponent for the best learning output pattern series corresponding to (a) MIN_1, and (b) MIN_2

desired outputs) should be examined to explore topological conjugacy between the best learned input pattern and the best learned output pattern. It is evident that f and d are topologically conjugate only if $Y(x)$ is homeomorphic. In this case, it can be directly considered that the best learned output follows a chaotic itinerary in the output space. If not, numerical analysis should be applied. In the current case, $Y(x) = x$, so none of the two methods is necessary.

6.3.3. Chaotic Response of the RMS Error

During additional training computations of the RMS error of the data set were made after each training epoch. Because of the significance of the RMS error, it is essential to know its fluctuation with respect to the number of the training epoch. In the case of the converging autoassosiative network, this fluctuation is very small. The RMS error covers a space of maximum range 0.000088 moving from 0.00043 to 0.000522. This tiny attractor, exhibits noticeable expanding properties. The evolution of the dominant Lyapunov exponent is shown in Figure 28. An Lcoef equal to 0.378 seems to be a good estimation for the representation of the dominant Lyapunov exponent of the RMS error attractor.

To show the significance of the chaotic properties of the RMS error, distributions of absolute error arising from testing the network after each training epoch should be taken in to consideration, and a correspondence between them and the RMS error (again calculated after training epochs) should be established. This is accomplished as follows. First, the RMS error of the data set

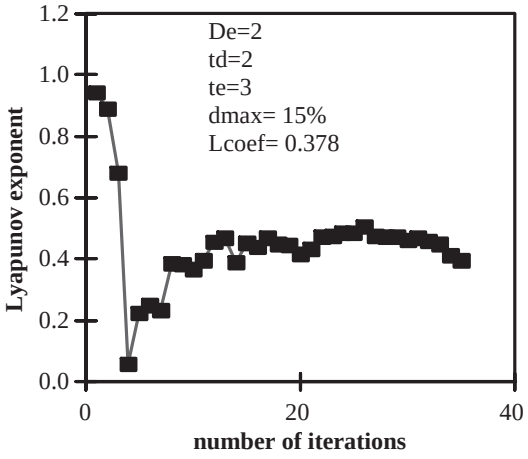

Figure 28: Evolution of the Lyapunov exponent for the RMS error during training

is calculated after the n_{th} training epoch. Then, the percentage of data in the data set, producing an output with absolute error exceeding a certain error level is calculated. Table 29 shows this correspondence for some values selected from the RMS data series.

AVERAGE ERROR	0.000369	0.000372	0.000377	0.000459
RMS ERROR	0.000434	0.000436	0.000437	0.000522
$\%ERR > 0.005$	0%	0%	0%	0%
$\%ERR > 0.001$	1%	1%	1%	0%
$\%ERR > 0.0005$	8%	12%	15%	24%

Table 29: RMS error in correspondence with absolute error distributions

As shown in the first three data columns, even a small fluctuation of 0.000001 causes an increase of 3-4% to the percentage of data, with absolute error greater than 0.0005. For the maximum fluctuation of the RMS error, the percentage of data with error larger than 0.0005 is about 24%, which is three times the value corresponding to the lower RMS error value as shown in column 4. It should be noted that, in this last case (of column 4), the absolute error remains under the 0.001 level for the whole data set, while in the other cases it is not.

7. Conclusions

Back Propagation neural networks with Functional Link inputs can successfully simulate the Logistic and Henon attractors. The general structure that produces the specific neural models is common for the two maps. The basic models are produced using single training. Their improvements are submitted to multiple training, a process from which a complete set of submodels is created. This set is successfully driven by an LVQ controller. The resulting decrease of RMS, Average and Absolute errors balances the high complexity of the whole system.

Absolute error series produced by neural models of chaotic attractors in a step by step presentation of a chaotic orbit during the recall phase, exhibit chaotic behavior. This directly affects the ability of the network to reconstruct the attractor as a set of points, or in its sequential form of orbits evolving in the phase space. The former can be considered feasible, while for the latter, only a local solution should be considered satisfactory.

Neural networks of simple structures, trained with ordinary values of training parameters using the Back Propagation algorithm, behave chaotically when pattern presentation during training is random. This may be expected from a non-converging network, but chaotic pattern competition seems to be present even when convergence is reached. It follows that the best estimations in the output are moving on a chaotic path as well. Consecutive computations of the RMS error of the input data set, during training, also form chaotic series. Although the RMS attractor is tiny, it seriously affects the fluctuation of the overall absolute error

References

[1] K. Aihara, T. Takabe, M. Toyoda M., Chaotic neural networks, *Physics Letters A*, **144**, No. 6, 7 (1990).

[2] T. Kathleen Alligood, Tim D. Sauer, James A. Yorke, *Chaos. An Introduction to Dynamical Systems*, Springer (1995)

[3] Gregory L. Baker, J. Gollub, *Chaotic Dynamics: An introduction*, Cambridge University Press (1990), 82-84.

[4] E.K. Blum, Xin Wang, Stability of fixed points and periodic orbits and bifurcations in analog neural networks, *Neural Networks*, **5** (1992), 577-587.

[5] Francois Chapeau-Blondeau, Gilbert Chauvet, Stable, oscillatory and chaotic regimes in the dynamics of small neural networks with delay, *Neural Networks*, **51** (1992), 735-743.

[6] Robert L. Devaney, *An Introduction to Chaotic Dynamical Systems*, Addison-Wesley Publishing Company, Inc. (1989).

[7] J.P. Eckman, S. Kamphorst, R.D. Oliffson, S. Ciliberto, Lyapunov exponents from time series, *Physical Review A*, **34**, No. 6 (1986).

[8] Mithel J. Feigenbaum, Universal behavior in nonlinear systems, *Physica 7D* (1983), 16-39.

[9] Ramazan Gencay, Nonlinear prediction of noisy time series with feedforward networks, *Physics Letters A*, **187** (1994), 397-403.

[10] Simon Haykin, *Neural Networks. A Comprehensive Foundation,* Macmillan College Publishing Company, Inc. (1994).

[11] M. Henon (1976), A two dimensional map with a strange attractor, *Commun. Math. Phys.*, No. 50 (1976), 69-77.

[12] Robert C. Hilborn, *Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers*, New York Oxford University Press (1994).

[13] Yeh I-Cheng, Modelling chaotic dynamical systems using extented neuron-networks, *Neural, Parallel and Scientific Computations*, **5** (1997), 429-438.

[14] A. Thomas Kepler, Sumeet Datt, Robert A. Meyer, L.F. Abbot, Chaos in a neural network circuit, *Physica D*, **46** (1990), 449-457.

[15] John F. Kolen, Jordan B. Pollack, Back Propagation is sensitive to initial conditions, *Complex Systems*, **4** (1990), 269-280.

[16] A. Lapedes, R. Farber, Nonlinear signal processing using neural networks: predictive and system modelling, *LA-UR-87-2662*, Los Alamos National Laboratory, Technical Report (1987).

[17] S.F. Masri, A.G. Chassiakos, T.K. Caughey, Structure - unknown nonlinear dynamic systems: identification through neural networks, *Smart Mater Struct.*, **1** (1992), 45-56.

[18] George J. Mpitsos, Robert M. Burton, Convergence and Divergence in neural networks: Processing of chaos and biological analogy, *Neural Networks*, **5** (1992), 605-625.

[19] H. Randons, G. Schuster, D. Werner, Fractal measures and diffusion as results of learning in neural networks, *Physics Letters A*, **174** (1993), 293-297.

[20] Georg Heinz Schuster, *Deterministic Chaos: An Introduction*, VCH, Germany (1995).

[21] Steve Renals, Richard Rohwer, A study of network dynamics, *Journal of Statistical Physics*, **58**, No. 5/6 (1990).

[22] F. Takens, In *Dynamical Systems and Turbulence*, **898** of *Lecture notes in Mathematics* (Ed-s. D.A. Rand, L.S. Young), Spinger, Verlag, Berlin (1981).

[23] Han L.J. Van der Maas, Paul F.M.J. Verschure, Peter C.M. Molenaar, A note on chaotic behavior in simple neural networks, *Neural Networks*, **3** (1990), 119-122.

[24] Paul F.M.J. Verschure, Chaos-based learning, *Complex Systems*, **5** (1991), 359-370.

[25] Allan Wolf, Jack B. Swift, Harry L. Swinney, John A. Vastano, Determining Lyapunov exponents from a time series, *Physica D*, **16** (1985), 285-317.

[26] Jon Wright, Method for calculating a Lyapunov exponent, *Physical Review A*, **29**, No. 5 (1984).